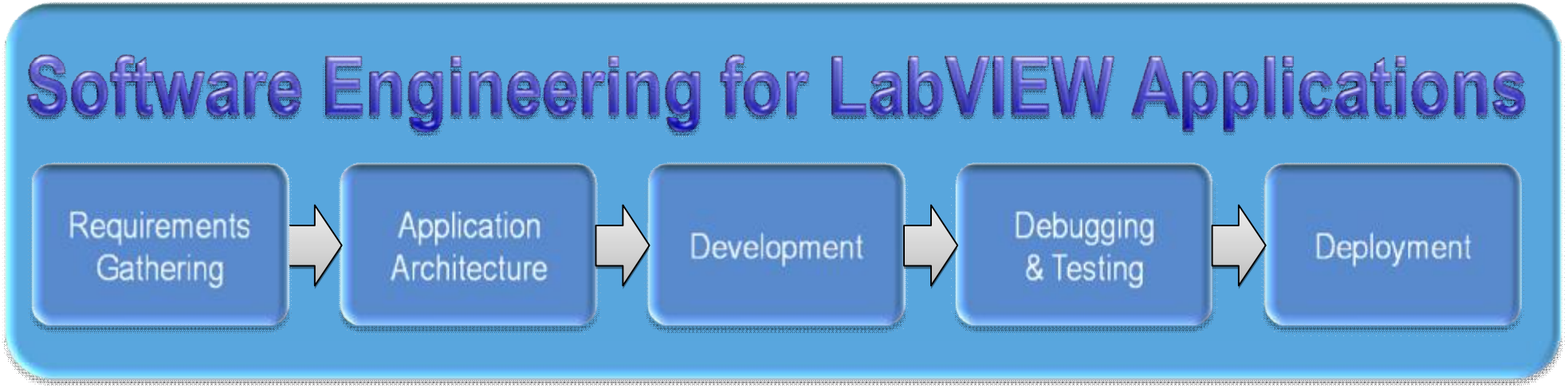
Software Engineering for LabVIEW Applications



TOOLS

Requirements Gateway	Design Patterns	Dataflow	VI Analyzer	Application Builder
	Object Orientation	MathScript	Real Time Execution Trace	Real Time
	Multicore	Statechart	Desktop Execution Trace	FPGA
		Simulation	Unit Test Framework	Embedded
		Express		

Automated Software Testing

Goals

1. Deliver a working product
2. Prove it works right
3. Mitigate risk of failure
4. Avoid last-minute changes

Why?

1. More complex software
2. Mission-critical applications
3. Team size is growing
4. Increased scrutiny
5. Decreased time

Software Quality Standards

Company wide certification standard

ISO 9000

Voluntary certification standard for consistent processes

CMMI

Process improvement model sponsored by the National Defense Industrial Association

Product specific certification for quality

DO-178B

FAA standard for avionics software

FDA 21 CFR Part 820

Medical device standard

National Instruments is ISO 9001 Certified (HW&SW)

KEMA

Affiliate with the N.V. KEMA in The Netherlands

CERTIFICATE

Certificate Number: 510312.000

The Quality System of

National Instruments
Headquarters:
11500 N. Mopac Expressway
Austin, TX 78759
United States

4031 Debrecen
1/A Hatar Str
Hungary

Including its implementation, meets the requirements of the standard:

ISO 9001:2000

Scope – Austin, TX:
Marketing, Design, Development, Manufacturing, Sales, Support for Software and Hardware Products for Personal Computers and Workstations that are used for Measurement and Automation

Scope – Debrecen, Hungary:
Production & distribution of software and hardware products for personal computers and workstations that are used for measurement and automation

This Certificate is valid until: August 31, 2010
This Certificate is valid as of: August 31, 2007
Certified for the first time: August 26, 1995

H. Pierre Sallé
President
KEMA-Registered Quality

The method of operation for quality certification is defined in the KEMA General Terms And Conditions For Quality And Environmental Management Systems Certifications. Integral publication of this certificate is allowed.

KEMA-Registered Quality, Inc.
4377 County Line Road
Chalfont, PA 18914
Ph: (215)997-4519
Fax: (215)997-3809

Accredited By:
ANAB

NI's QSM/NPI/SEP

Quality Management System (QMS)

- Covers all of NI

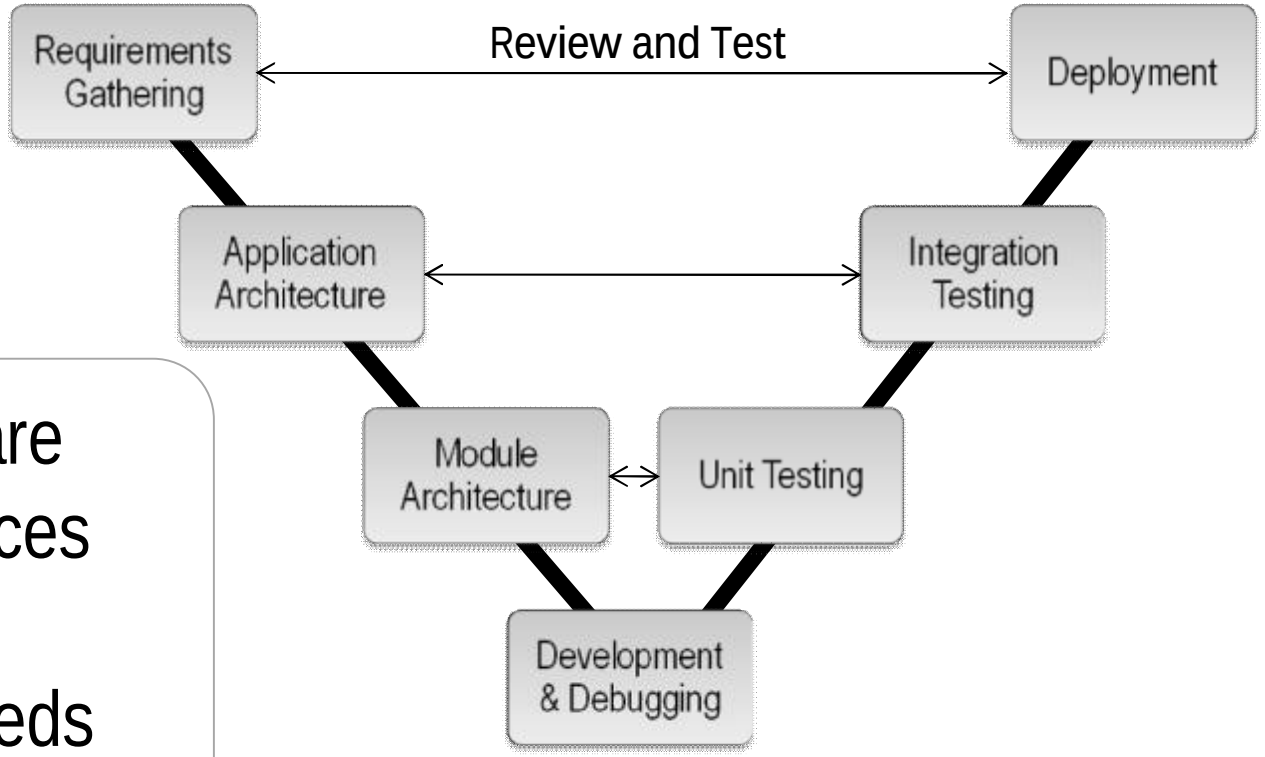
New Product Introduction (NPI) process

- Engineering/Marketing/Manufacturing/Sales

Software Engineering Process (SEP)

- Engineering only
- How SW Engineers and Project Managers execute NPI for SW products

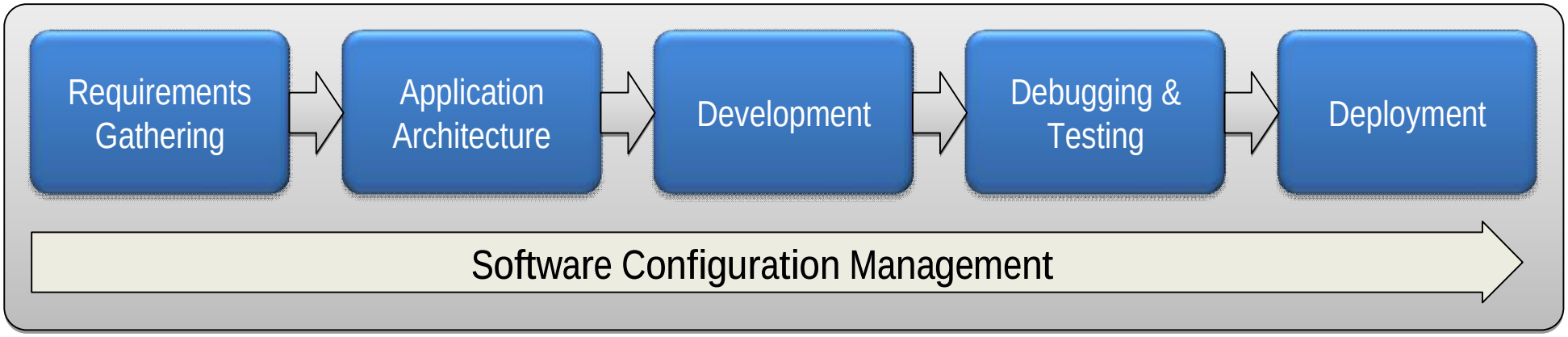
Software Engineering Process Models



A model for software development practices

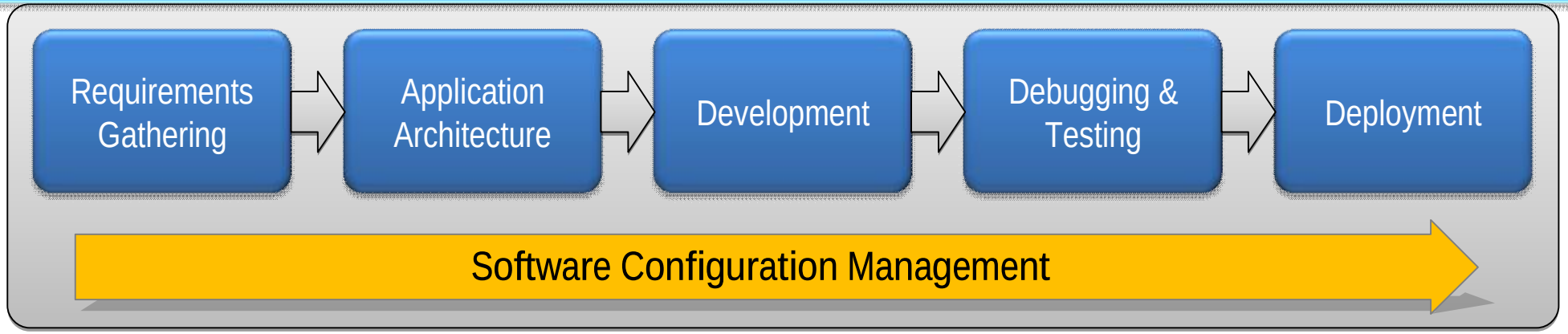
Prescribes what needs to be done before moving on

Software Engineering Process



- Process is independent of programming language
- Demonstrate a particular process for certification
- Automate this process for LabVIEW with toolkits and add-ons
- SCM is applied throughout process

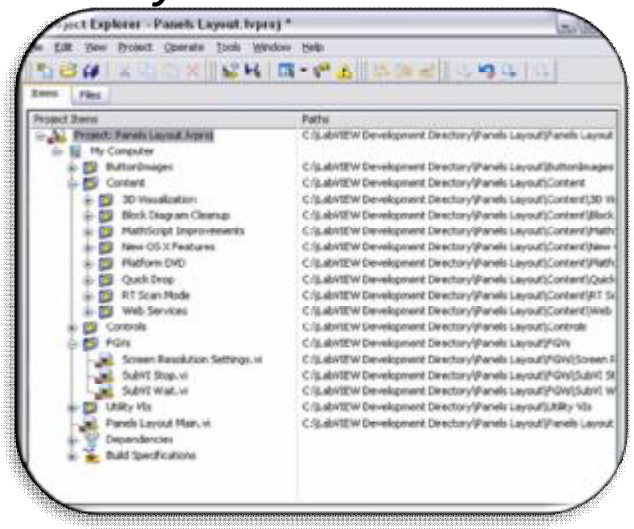
Configuration Management Challenges



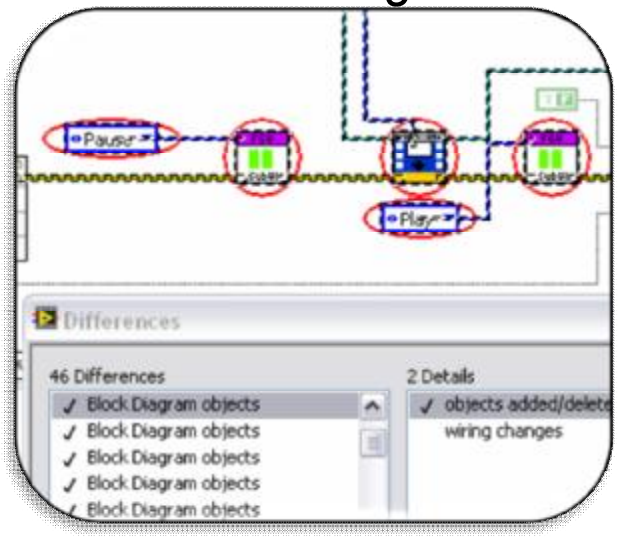
- Defining a central repository of code
- Management of multiple developers
- Detection and resolution of code collisions
- Tracking behavioral changes
- Identification of changes are who made them
- Ensuring everyone has latest copy of code
- Backing up old code versions
- Managing *all* files, not just source code

Software Configuration Manager for LabVIEW

System Level View

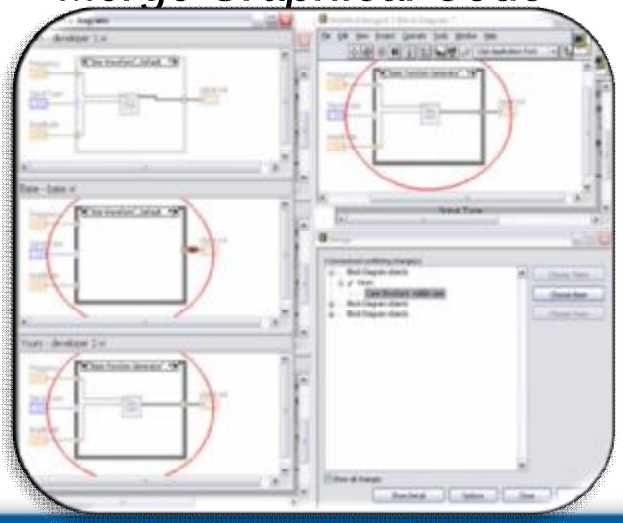


Track Changes

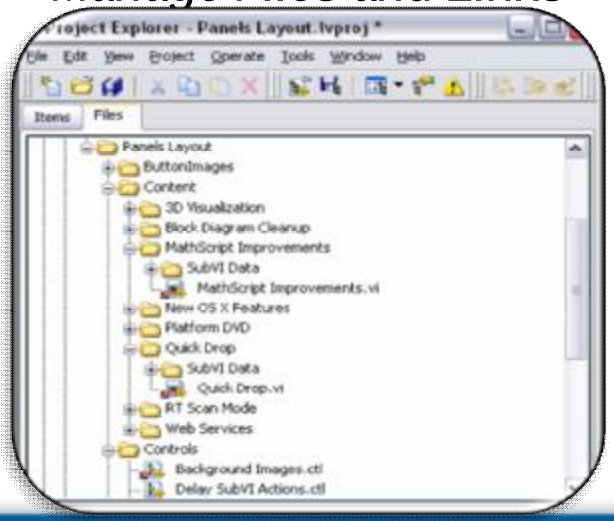


Integrate with SCC

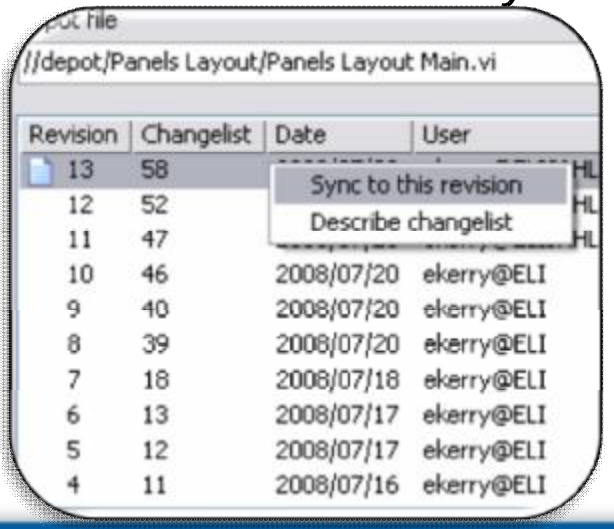
Merge Graphical Code



Manage Files and Links

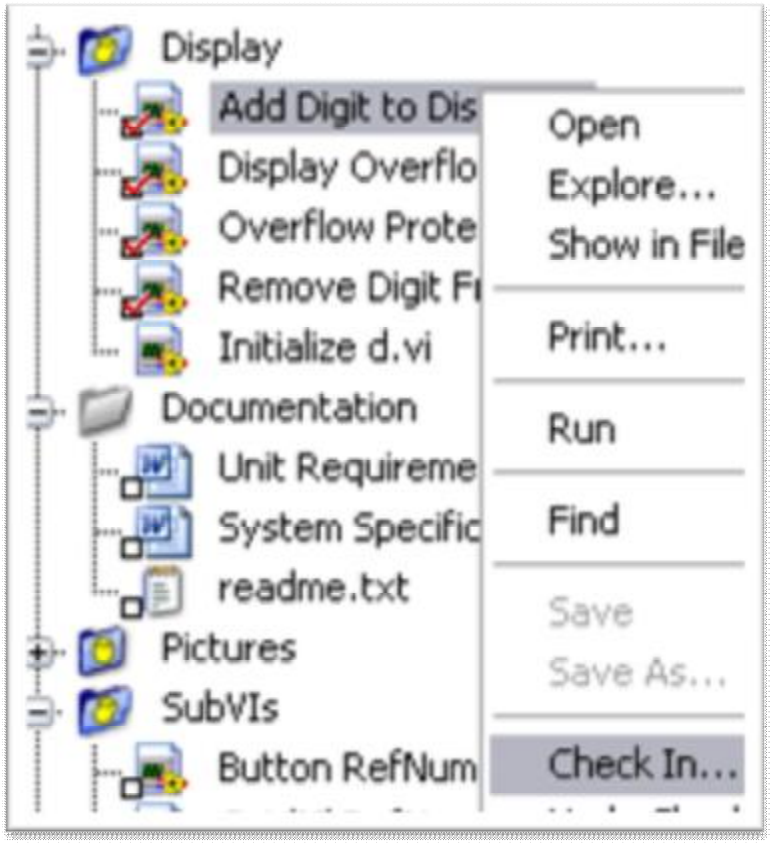


View Revision History



Source Code Control Integration with LabVIEW

- Third-party source control integration with:
  - Microsoft Visual SourceSafe
  - Microsoft Team System
  - Perforce
  - Rational ClearCase
  - PCVS (Serena) Version Manager
  - MKS Source Integrity
  - Seapine Surround SCM
  - Borland StarTeam
  - Telelogic Synergy
  - ionForge Evolution
  - subVersion**
- Access SCC tools via LabVIEW Project
- Project specific settings*



**New in LabVIEW 8.5*

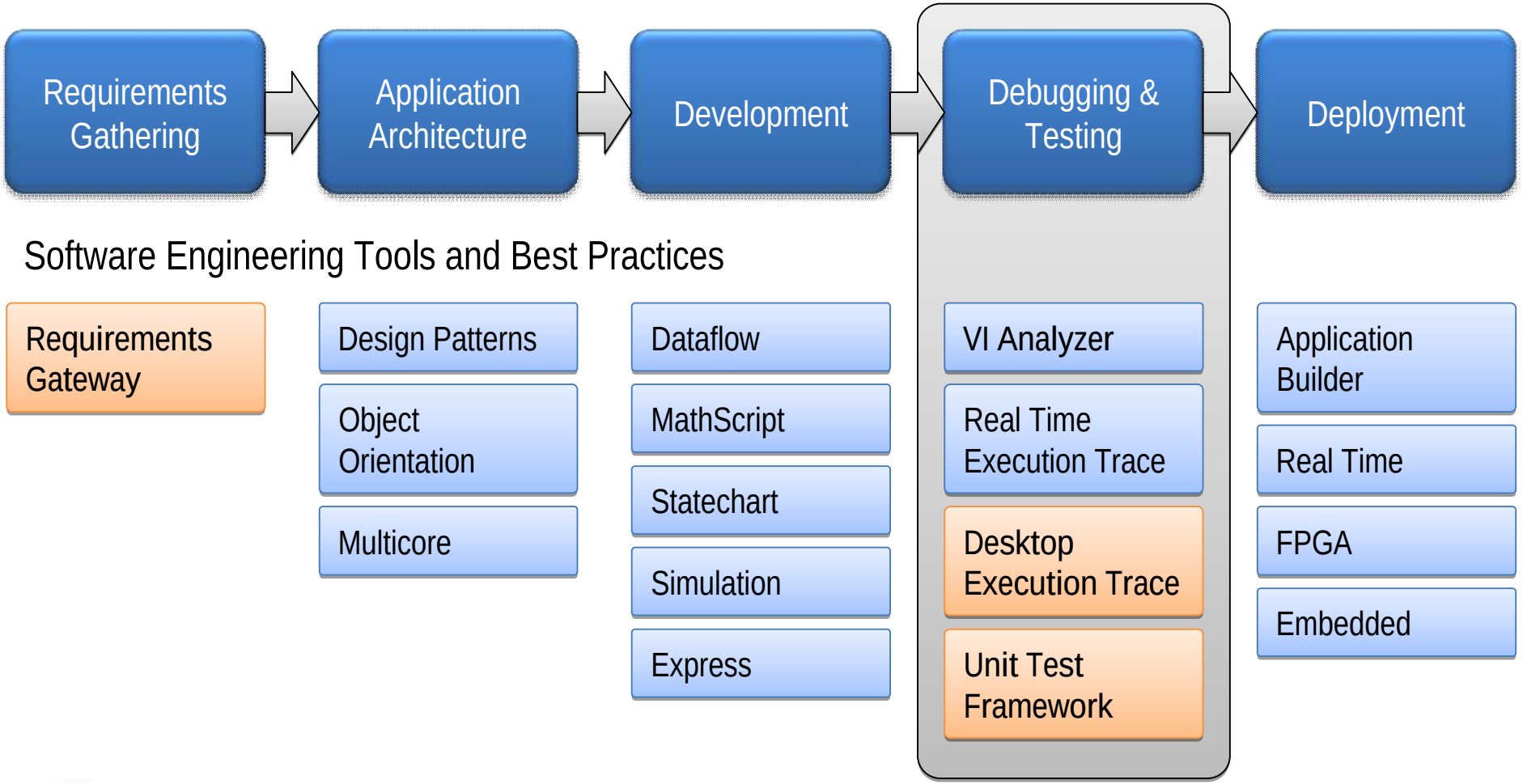
**subVersion is open source and requires plug-in

The Software Engineering Process TOOLS

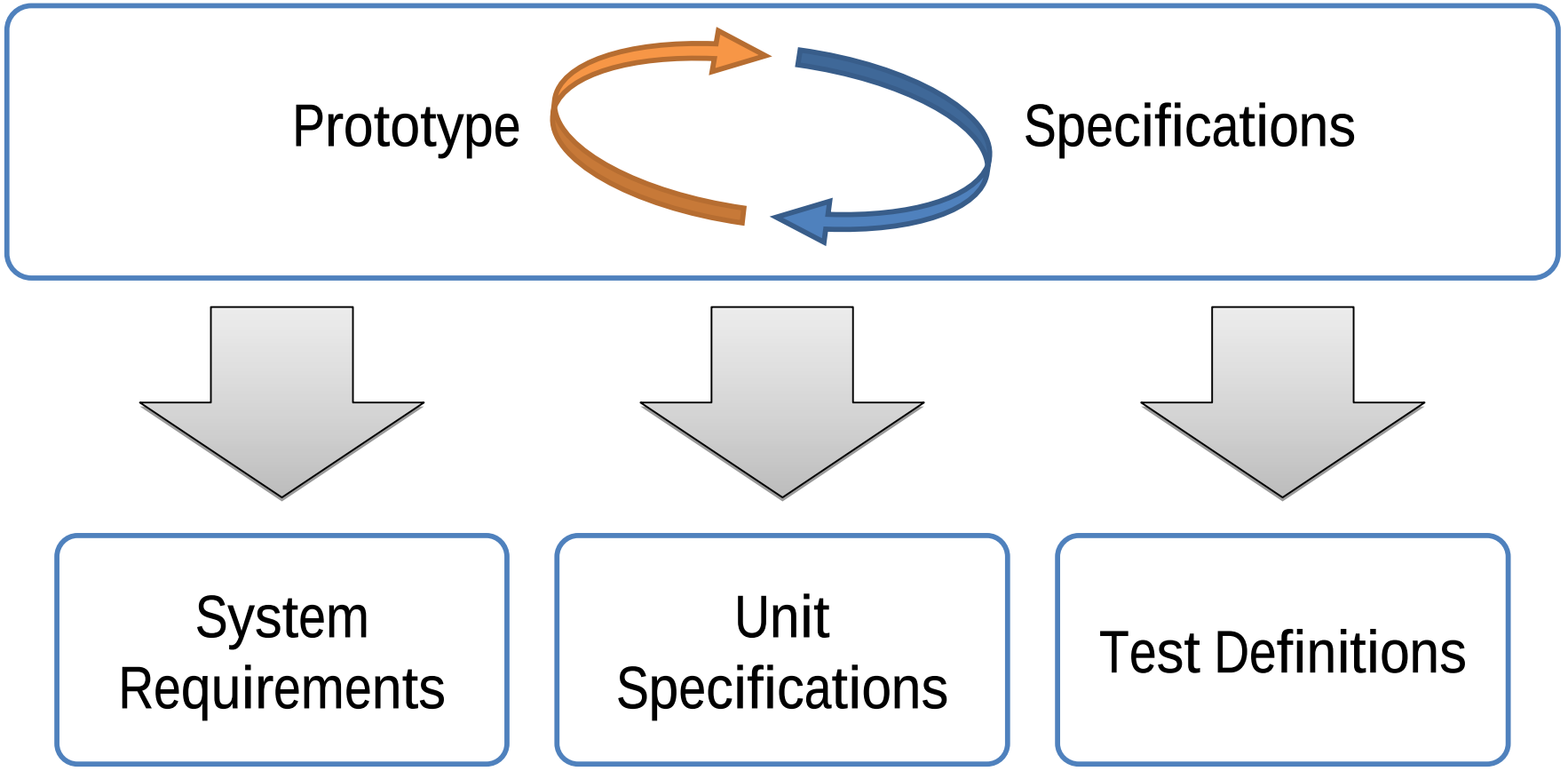
Software Engineering Tools and Best Practices

Requirements Gateway	Design Patterns	Dataflow	VI Analyzer	Application Builder
	Object Orientation	MathScript	Real Time Execution Trace	Real Time
	Multicore	Statechart	Desktop Execution Trace	FPGA
		Simulation	Unit Test Framework	Embedded
		Express		

Source Code Control Integration with LabVIEW



Software Design Phase

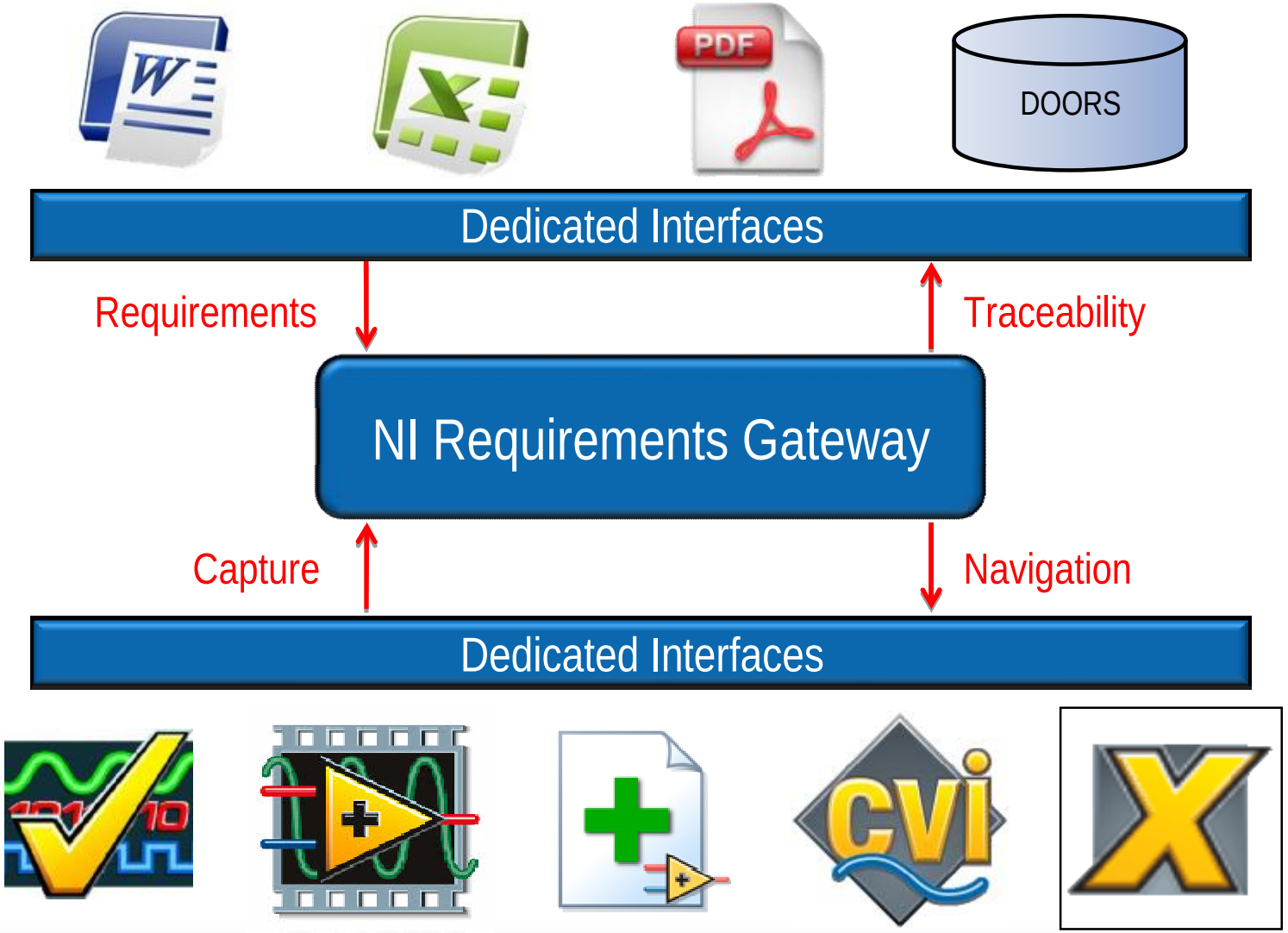


Requirements Gathering Challenges

- Agreeing upon what the application will do
- Defining acceptable and safe behaviors
- Defining tests for individual components
- Traceability from code to documents

Upstream	Text	Downstream
REQ_ControlLevel	The application must maintain a separate control algorithm for controlling tank levels.	Level.vi
REQ_ControlTemperature	The application must maintain a separate control algorithm for controlling temperature.	Temperature.vi
REQ_DisplayGraphInflow	The display must display a graph indicating the inflow rate.	Inflow Rate History
REQ_DisplayGraphLevel	The display must contains a graph indicating the tank level.	Tank Level History
REQ_DisplayGraphTemperature	The display must contains a graph indicating the tank temperature.	Tank Temperature History
REQ_DisplayHeatIndicator	The display must show whether furnace is on or off.	Heat Indicator
REQ_DisplayMinMaxLevels	The display must allow the operator to control the minimum and maximum levels for the tank.	Low Level Limit (lbm)
REQ_DisplayMinMaxLevels	The display must allow the operator to control the minimum and maximum levels for the tank.	High Level Limit (lbm)
REQ_DisplayMinMaxTemp	The display must allow the operator to control the minimum and maximum levels for the tank temperature.	Low Temp Limit (deg F)

Requirements Traceability Solution from NI

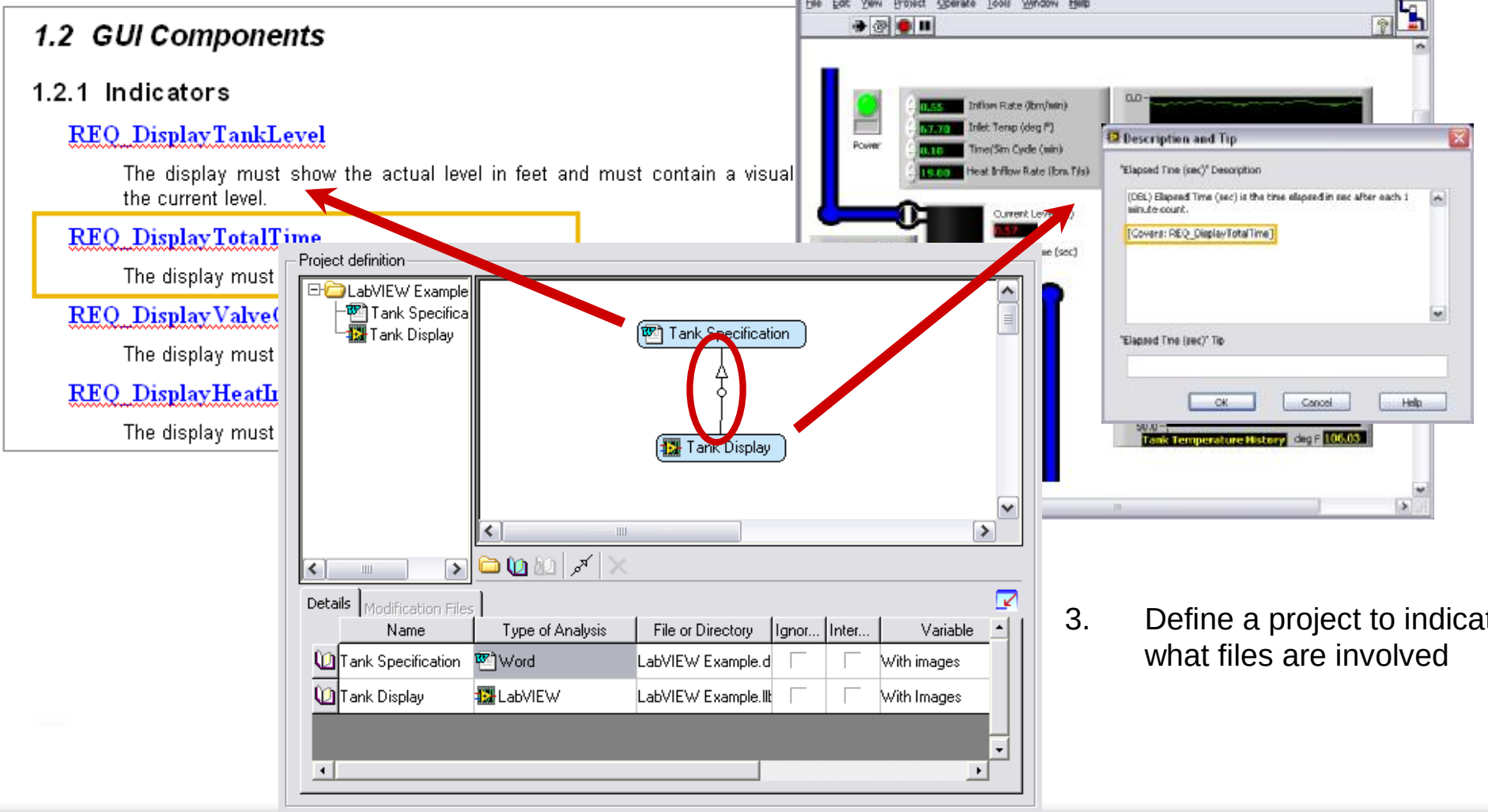


Requirements Coverage with LabVIEW

1. Requirements defined in Word, Excel, PDF, DOORS, RequisitePro, …

2. LabVIEW references requirements

3. Define a project to indicate what files are involved

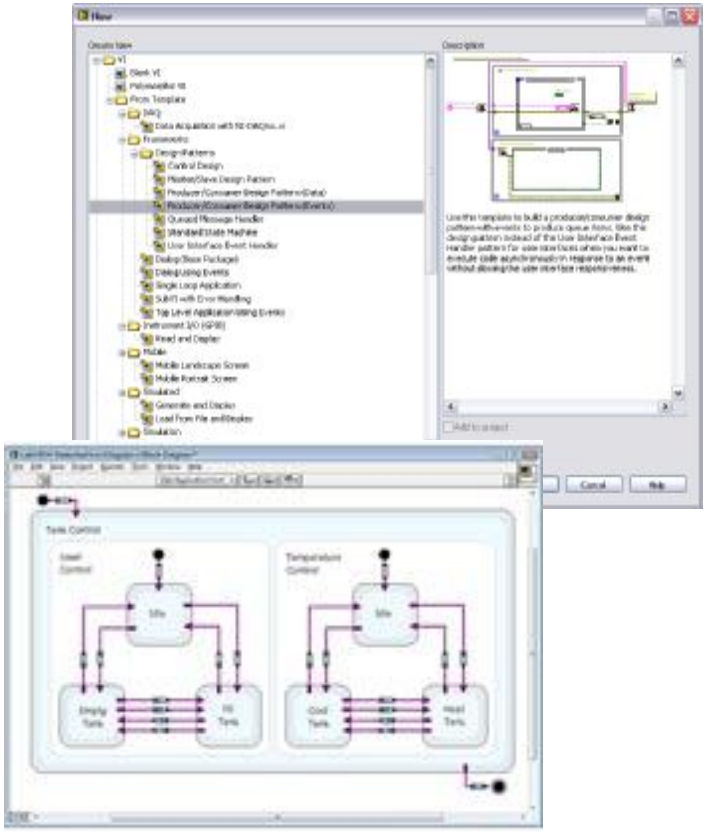


Application Architecture Goals

- Determine how code will be divided based upon functionality
- Develop standard interfaces and APIs for code modules
- Ensure code scales and is readable and maintainable

Application Architecture TOOLS

- LabVIEW Design Patterns
- LabVIEW Statechart Module
- UML Tools
- Object-Oriented Design

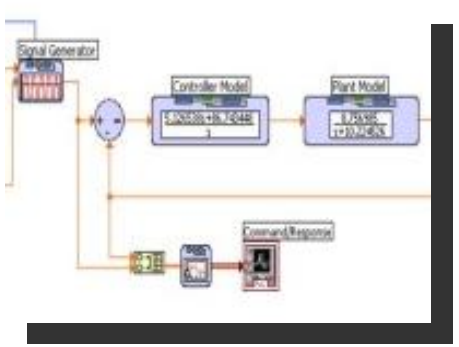


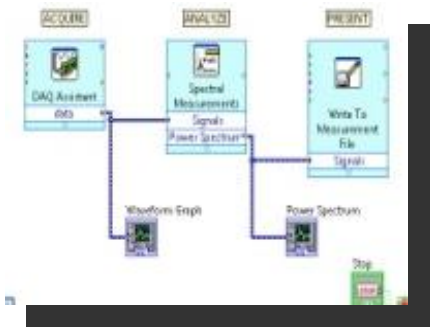
Development Goals

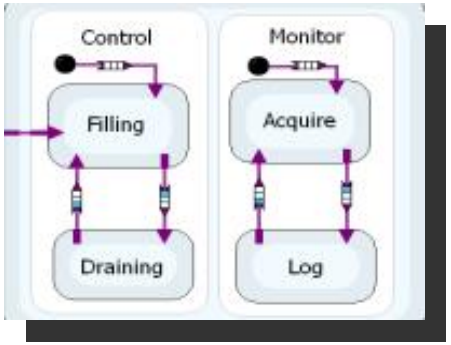
- Implement the application
- Fulfill requirements documentation
- Detect and resolve incorrect code behavior
- Functionality passes test parameters
- Abide by style guidelines
- Optimize performance and execution

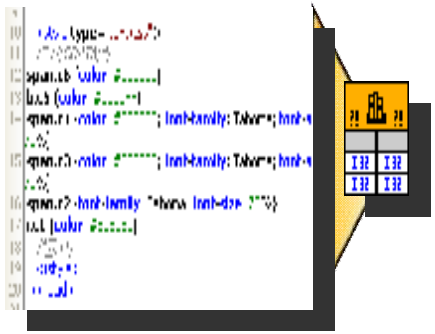
Develop Using Multiple Models of Computation

Requirements Gathering → Application Architecture → **Development** → Debugging & Testing → Deployment



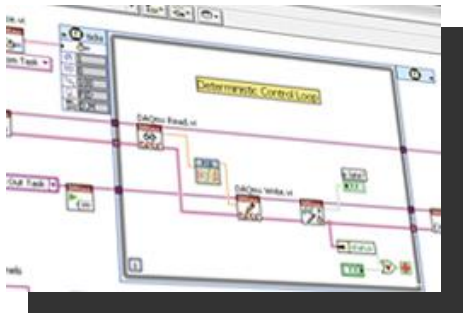






```
1 c = 0.285 + 0.013i;
2 [X Y] = meshgrid(x, y);
3 z = X + i*Y;
4 for k=1:30
5     z = z.^2 + c;
6 end
```

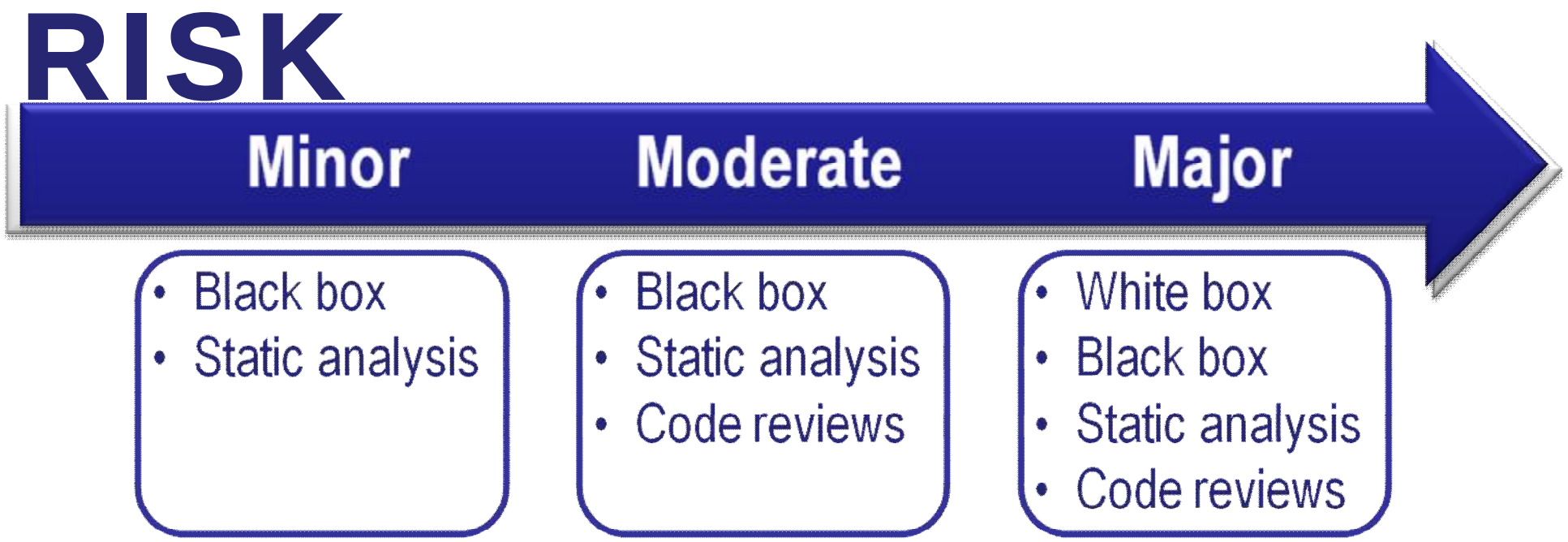
x
y
z



Debugging & Testing Practices

- Iterative testing during development
- Manual peer code review
- Static and dynamic code analysis
- Validation against requirements

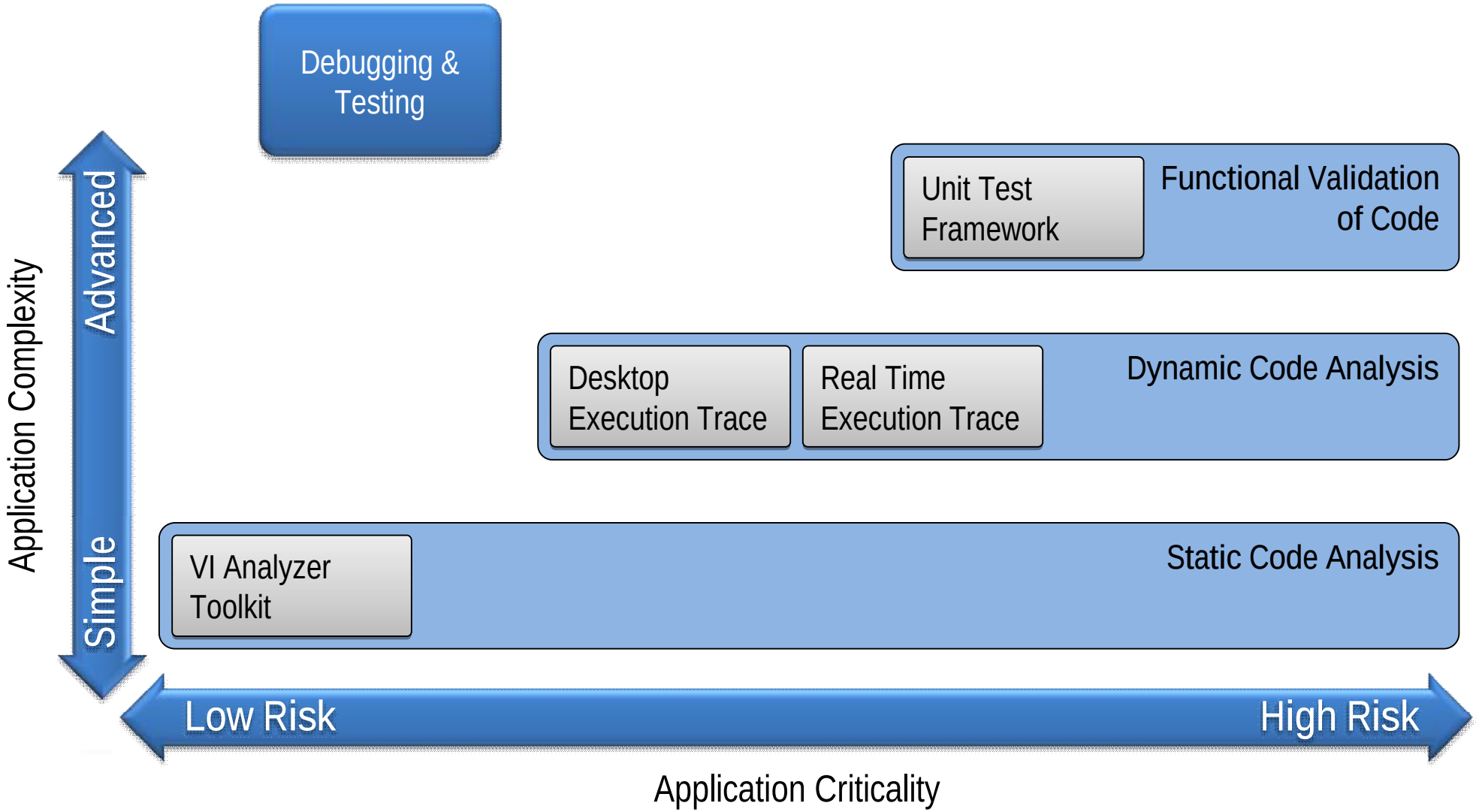
Test Plan Development



Debugging & Testing TOOLS

- Constantly compiling
- Highlight execution
- Breakpoint manager
- Diagram cleanup
- National Instruments Style Guideline
- VI Profiler
- Coercion (Memory) Dots
- Conditional disable structures
- VI Analyzer Toolkit
- RT Execution Trace Toolkit
- Desktop Execution Trace

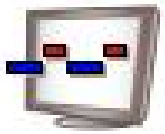
Debugging & Testing TOOLS



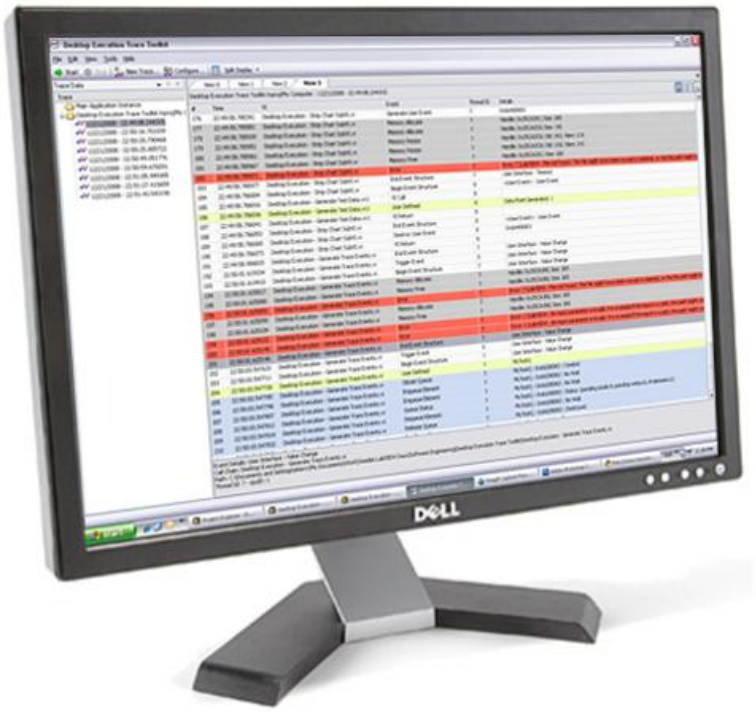
Goals of Dynamic Code Analysis

- What is consuming system memory?
- Am I capturing all the errors in my application?
- What was the last event to occur before…?
- What was the call-chain that led us to…?
- What thread is it executing in?
- Am I actually entering a specific event-case?
- What happened inside a structure?
- What order to these events occur in?
- Is a daemon process running in the background?
- Does the code behave different in an executable?

Desktop Execution Trace Toolkit



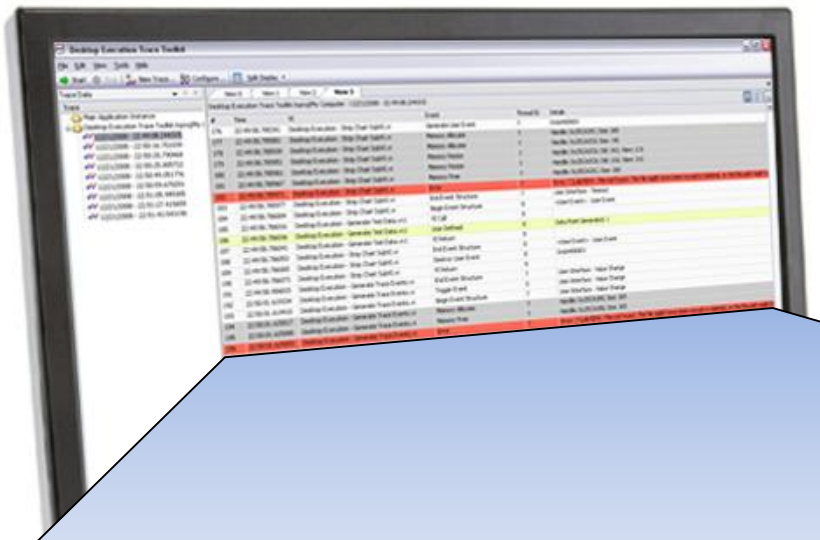
Monitor the execution of LabVIEW code at run-time in order to debug common problems in large applications such as memory leaks and un-handled errors.



Trace During Run-Time:

- Event Structures
- Memory Allocation
- Queues / Notifiers
- Reference Leaks
- Thread ID
- Unhandled Errors
- Dynamic / Static SubVIs
- Custom User Strings

Desktop Execution Trace Toolkit



Trace During Run-Time:

- Event Structures
- Memory Allocation
- Queues / Notifiers
- ference Leaks

Strip Chart SubVI.vi	Memory Resize	7	Handle: 0x25CA3C8; Old: 142; New: 118
Strip Chart SubVI.vi	Memory Allocate	7	Handle: 0x25CA3C8; Size: 142
Strip Chart SubVI.vi	Memory Resize	7	Handle: 0x25CA3C8; Old: 142; New: 118
Strip Chart SubVI.vi	Error	7	Error: 7 (LabVIEW: File not found. The file might have
Generate Trace Events.vi	User Defined	7	MyTestQ
Generate Trace Events.vi	Obtain Queue	7	MyTestQ - 0x66200002 : Created
Generate Trace Events.vi	Enqueue Element	7	MyTestQ - 0x66200002 : No Wait

Trace Production Systems Remotely

LabVIEW Desktop Execution Trace Toolkit

Run-Time Execution Information

Network

VIs and Debuggable Executables

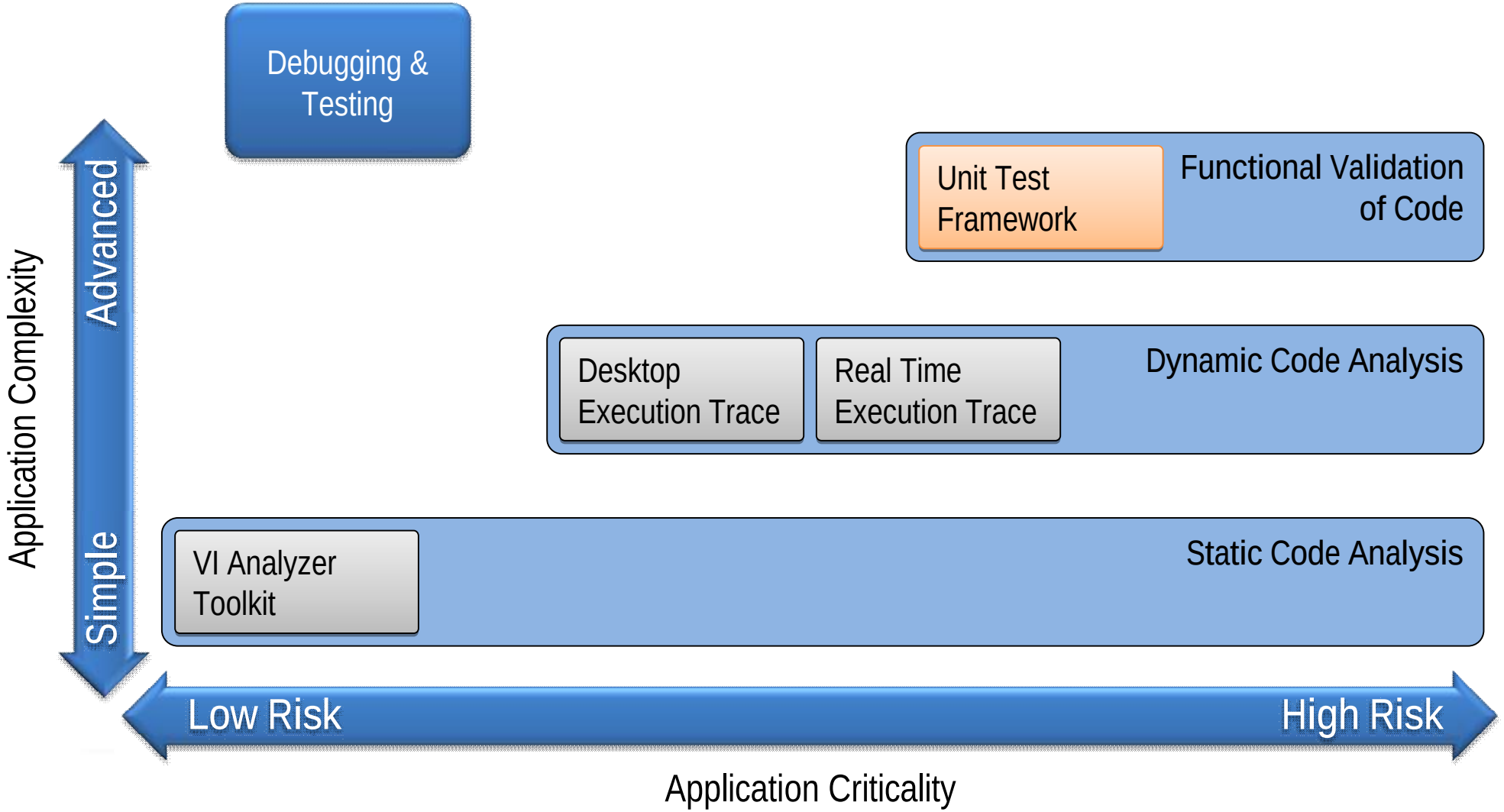
LabVIEW Real-Time Execution Trace Toolkit

Run-Time Execution Information

Network

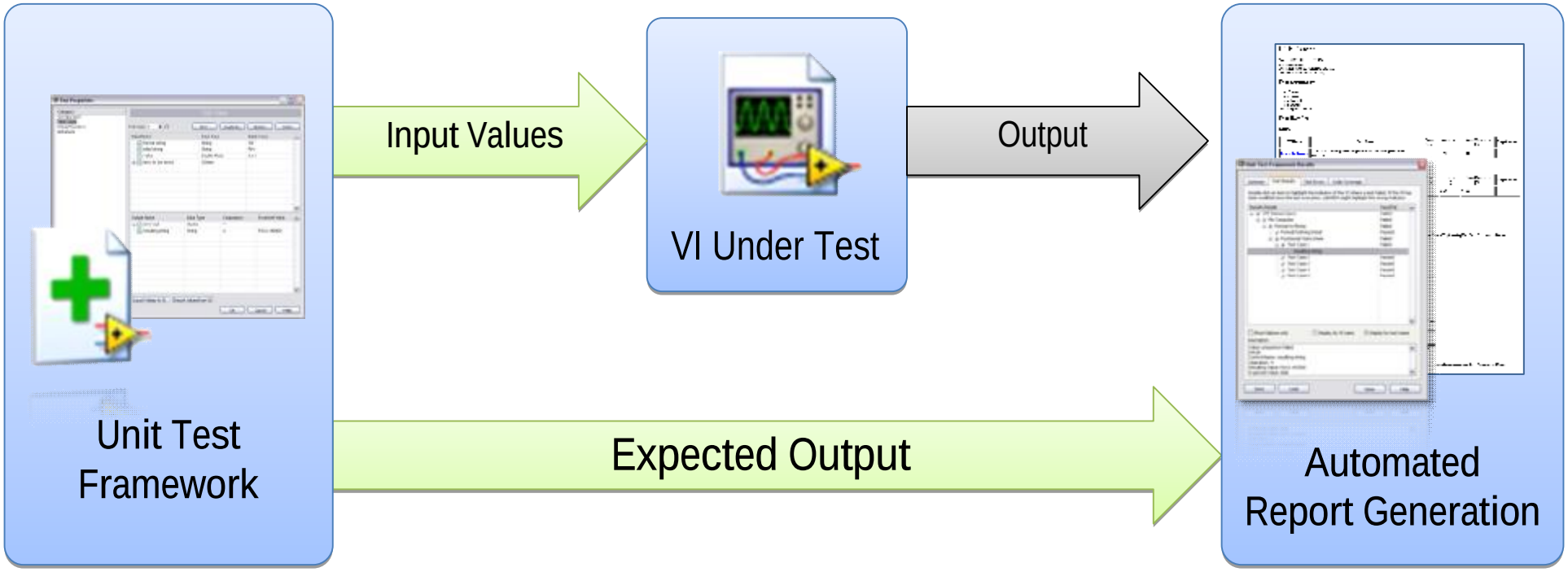
Deployed Real-Time Applications

Tools for Debugging and Testing



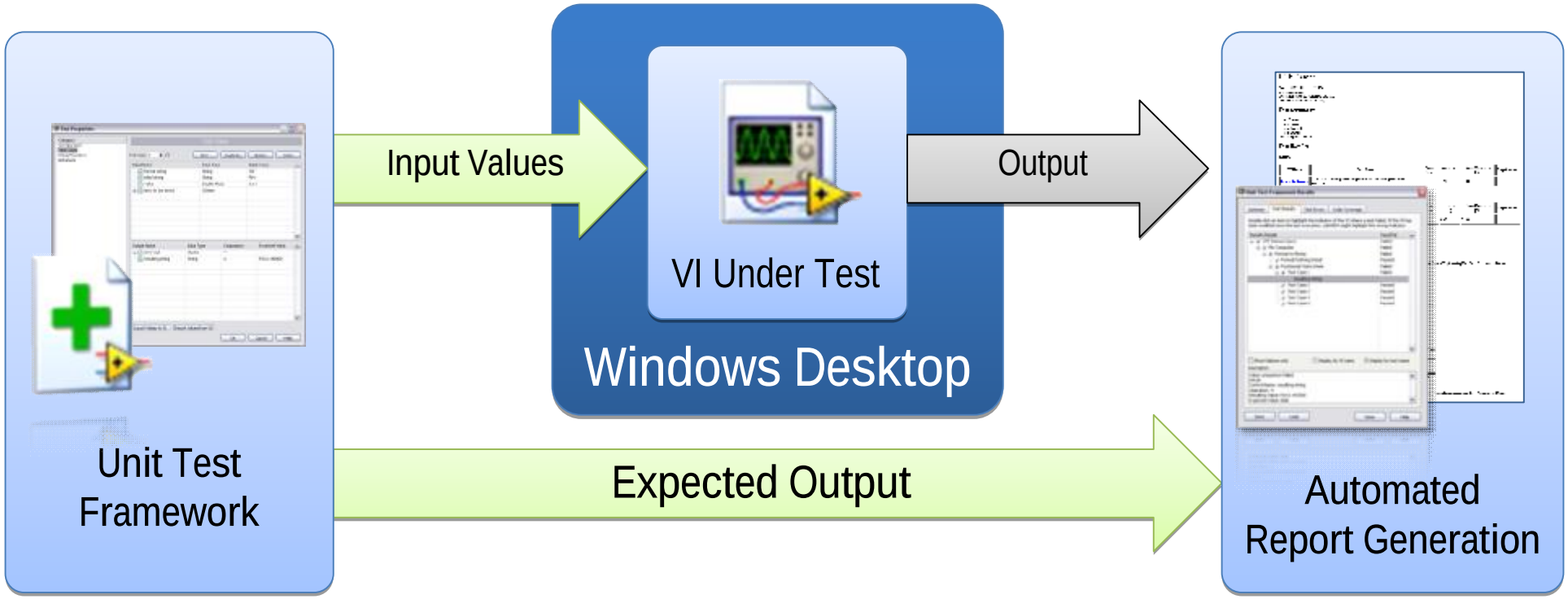
LabVIEW Unit Test Framework

Automatically test and validate VIs against requirements in order to demonstrate that it is working correctly according to design documents.



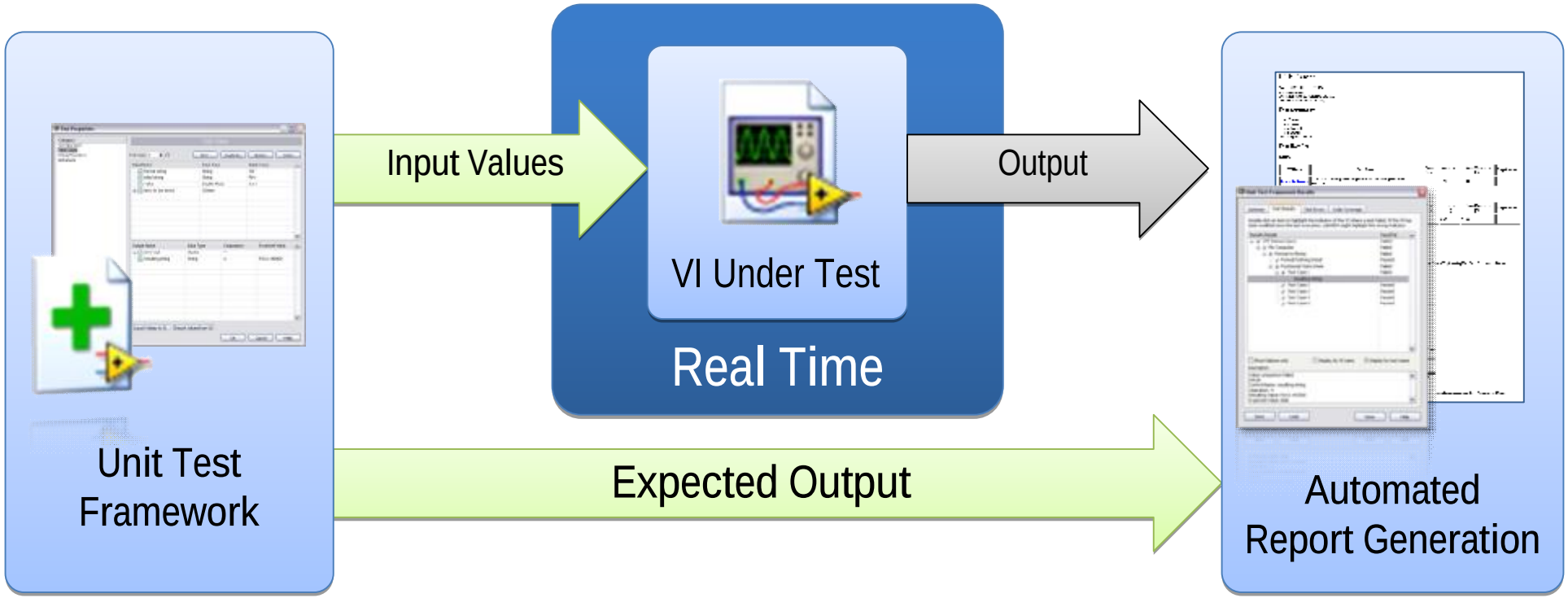
Test vector = Input value(s) + Expected output(s)

LabVIEW Unit Test Framework



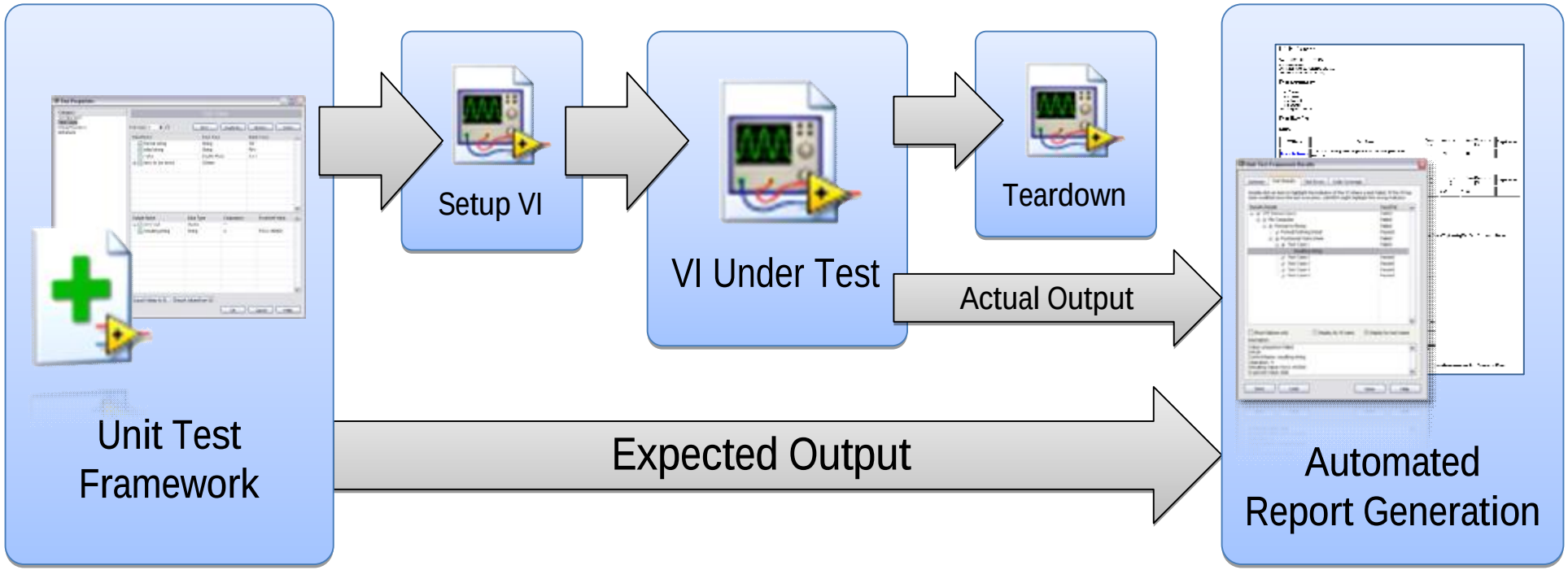
Test vector = Input value(s) + Expected output(s)

LabVIEW Unit Test Framework

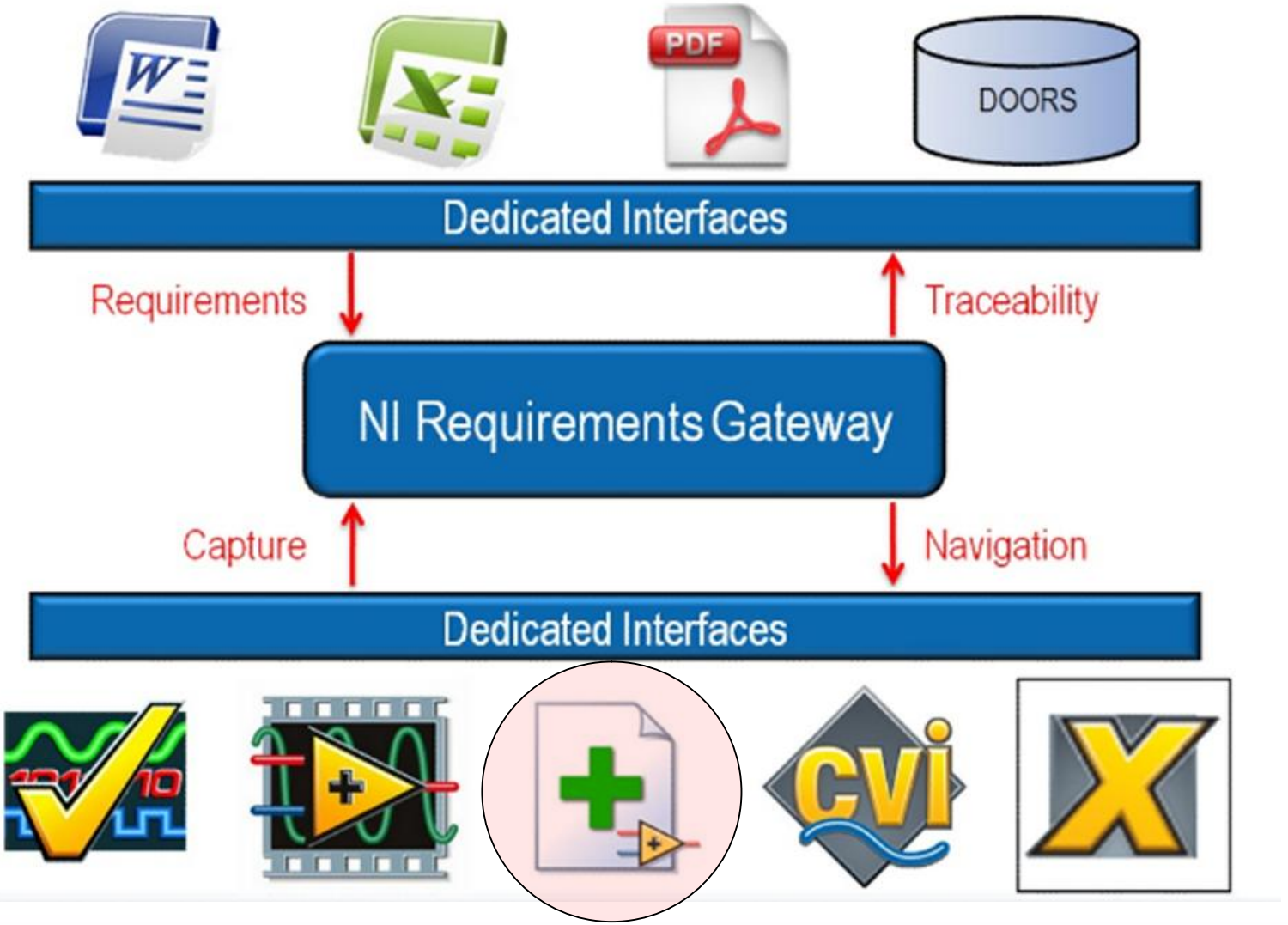


Test vector = Input value(s) + Expected output(s)

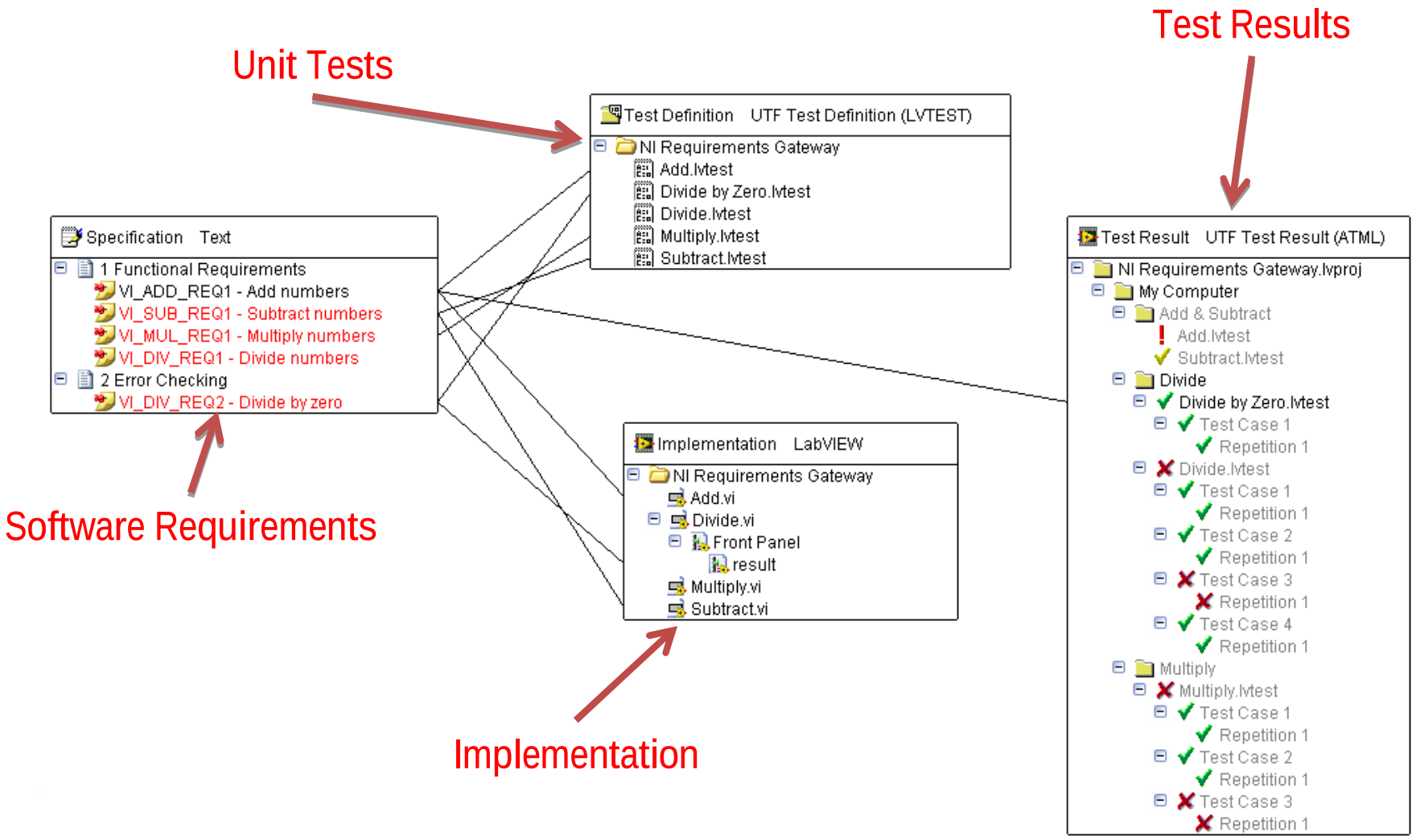
LabVIEW Unit Test Framework



Requirements Traceability Solutions



Integration with Requirements Gateway



Deployment Goals

- Determine repeatable process
- Deploy to determine behavior on other targets
- Deliver a product to customer or end-user
- Make every installer and driver required for operation easily portable
- Remove information about sensitive code

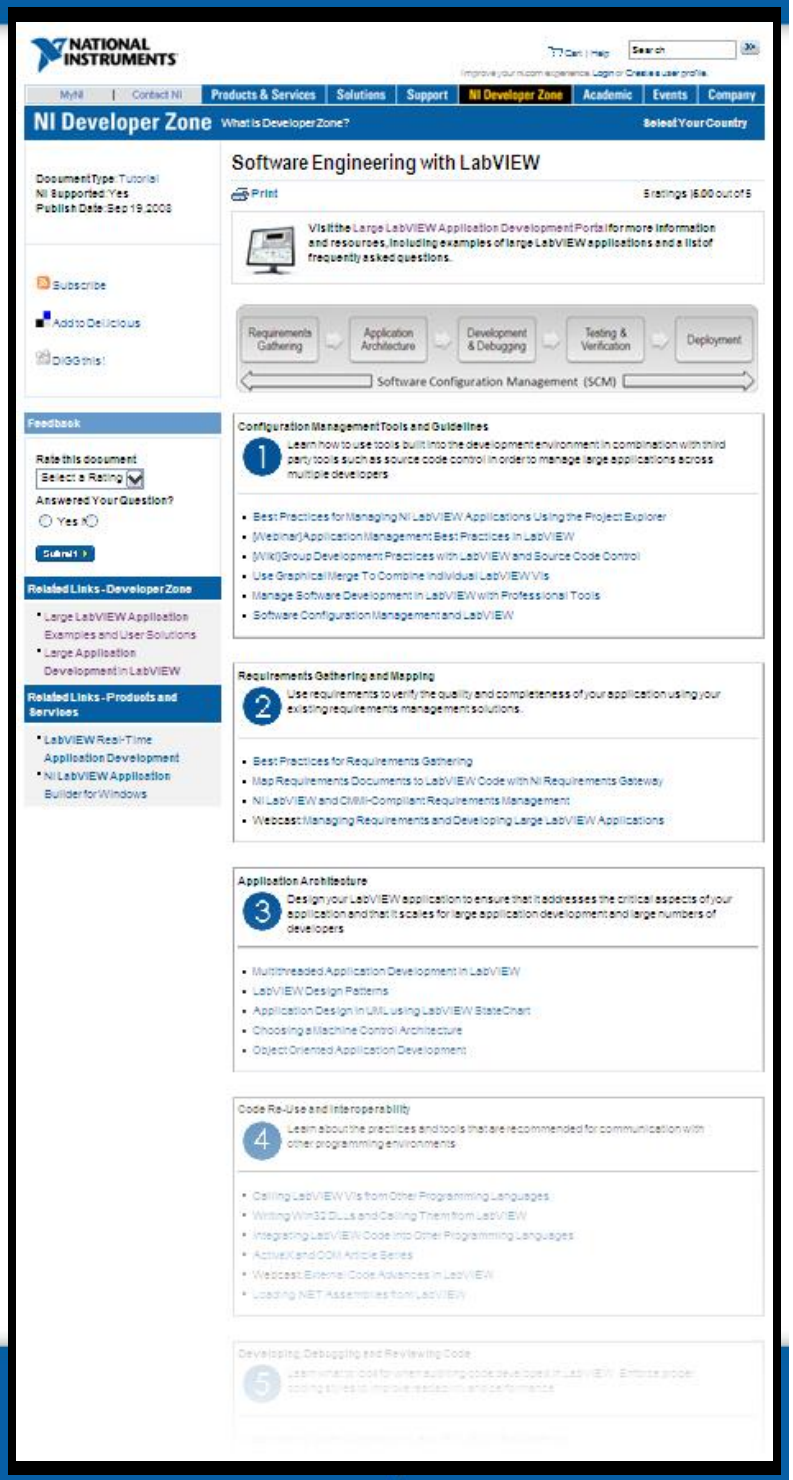
Deployment Tools

- LabVIEW Applications Builder
- LabVIEW Project
  - Many deployment documents on www.ni.com
- Replication tool for LabVIEW Real-Time
  - Norton Ghost, etc for Windows

Deploy Software to Embedded Hardware Targets

NI LabVIEW Application Builder

ni.com/labview/power



Learn more about:

Software Engineering
Configuration Management
Development Practices
Tools for Validation and Testing